Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A processing core, comprising:

a first source register including a plurality of first operands, wherein the source address of each first operand is identified in a single instruction;

a plurality of second operands, wherein:

the plurality of second operands are equal in value to an immediate value, and

the immediate value is specified in the single instruction;

a prescaler which scales the plurality of operands according to a predetermined scaling factor which is divisable by two

a bitwise inverter coupled to at least one of the first plurality of operands and the second plurality of operands;

a destination register including a plurality of results;

a plurality of arithmetic processors respectively coupled to the first operands, second operands and results, wherein each arithmetic processor computes one of a sum and a difference of the first operand and a respective second operand, and wherein the computation operation to be undertaken by each arithmetic processor is specified in the single instruction.

2. (Original) The processing core of claim 1, further comprising an integrated circuit which includes the first source register, destination register and arithmetic processor.

3. (Canceled)

4. (Previously Presented) The processing core of claim 1, wherein:

each arithmetic processor computes at least one of:

the result of the first operand plus another operand plus the immediate value; and

the result of the first operand minus another operand minus the immediate value; and

each of the first operand, the another operand and the immediate value are represented with a plurality of bits.

5. (Previously Presented) The processing core of claim 1, wherein the immediate value is signed.

6. (Canceled)

7. (Original) The processing core of claim 1, wherein a first width of the first source register is a positive integer multiple of a second width of the first operand.

8. (Previously Presented) The processing core of claim 1, wherein the sum and the difference are performed on a same carry look-ahead adder.

9. (Previously Presented) A method for performing arithmetic processing, the method comprising the steps of:

loading a first and second operands from a primary source register;

loading a third and fourth operands, wherein:

the third and fourth operands are an immediate value specified in an the single instruction, and

the third and fourth operands are equal in value;

scaling the third and fourth operands according to a predetermined scaling factor which is divisable by two;

performing an arithmetic function on the first and third operands to produce a first result;

performing the arithmetic function on the second and fourth operands to produce a second result; and

storing the first and second results in a destination register.

10. (Original) The method for performing arithmetic processing of claim 9, further comprising a step of inverting the third and fourth operands.

11. (Original) The method for performing arithmetic processing of claim 9, further comprising a step of adjusting at least one of the first and second results to avoid saturation of the destination register.

12. (Previously Presented) The method for performing arithmetic processing of claim 9, wherein the step of performing an arithmetic function on the first and third operands comprises calculating the first operand plus the third operand plus a positive integer, wherein each of the first operand, the third operand and the positive integer are each represented with a plurality of bits.

13. (Previously Presented) The method for performing arithmetic processing of claim 9, wherein the step of performing an arithmetic function on the second and fourth operands comprises calculating the second operand minus the fourth operand minus a positive integer, wherein each of the first operand, the third operand and the positive integer are each represented with a plurality of bits.

14 - 15. (Canceled)

16. (Original) The method for performing arithmetic processing of claim 9, wherein the two performing steps are performed, at least partially, coextensive in time.

17. (Previously Presented) The method for performing arithmetic processing of claim 9, wherein the two performing steps use a ripple look-ahead adder.

18. (Currently Amended) A method for performing arithmetic processing, comprising the steps of:

receiving a single instruction comprising an arithmetic function to be performed using an immediate value, and operands from first and second source addresses in a primary source register, wherein the immediate value and the first and second source addresses are specified in the single instruction;

loading a first and second operands respectively from the first and second addresses in the primary source register;

loading **a third and fourth operand each comprising** the immediate value;

scaling the third and the fourth operand according to a predetermined scaling factor which is divisable by two;

performing the arithmetic function on the first operand and **the third operand** ~~**immediate value**~~ to produce a first result;

performing the arithmetic function on the second operand and **the fourth operand** ~~**immediate value**~~ to produce a second result; and

storing the first and second results in a destination register.

19. (Previously Presented) The method for performing arithmetic processing of claim 18, wherein the immediate value is comprised of nine bits.

20. (Previously Presented) The method for performing arithmetic processing of claim 18, wherein the immediate value is comprised of thirteen bits.

21. (Original) The method for performing arithmetic processing of claim 18, wherein the two performing steps are performed, at least partially, coextensive in time.

22. (Original) The method for performing arithmetic processing of claim 18, further comprising a step of adjusting at least one of the first and second results to avoid saturation of the destination register.

23. (Previously Presented) The processing core of claim 1, wherein the instruction is a very long instruction word (VLIW) instruction.

24. (Previously Presented) The method for performing arithmetic processing of claim 9, wherein the steps of the method are initiated by a single instruction issue.